

Computer Science

K–12

Section 05

Computer Science K–12

1 Knowledge of problem solving and algorithms

1. Distinguish between object-oriented and procedural programming paradigms.
2. Identify the stages of the software development process (i.e., problem definition, analysis, design, implementation, testing, maintenance).
3. Identify an appropriate algorithm for a given problem.
4. Trace an algorithm and predict outputs for a given input.
5. Identify a minimum set of data necessary for testing a computer solution.
6. Identify problems appropriate for computer solution.
7. Distinguish between the classes of algorithmic constructs (i.e., sequence, decision, iteration).
8. Identify appropriate and efficient search algorithms for linear structures (i.e., sequential and binary).
9. Identify appropriate and efficient structures for searchable data (i.e., linear lists, binary search trees, hash tables).
10. Identify appropriate and efficient sorting algorithms for data sets (e.g., selection, insertion, merge, quick sort).
11. Identify string-processing algorithms (e.g., concatenation, substring extraction, pattern matching).

2 Knowledge of data types and structures

1. Distinguish between local and global identifiers in a procedural program.
2. Distinguish between constants and variables.
3. Distinguish between integer, floating point, character, Boolean, and object data types.
4. Distinguish between data structures or types (e.g., arrays, strings, linked lists, trees, hash tables, records, files, stacks, queues, sets, maps).
5. Distinguish between instance, class, and local (method) variables in an object-oriented program.
6. Identify components of class declarations for an object-oriented program (i.e., instance and class variables, constructors, methods).

7. Distinguish between public and private methods in an object-oriented program.
8. Identify key features of object-oriented programs (i.e., encapsulation, inheritance, polymorphism).

3 Knowledge of computer programming (All programming will be done in Logo, Visual Basic, C++, and Java)

1. Predict the output of a given program containing sequential, conditional, or iteration statements.
2. Complete a program segment involving only sequential execution when given an incomplete program with a specified output.
3. Complete a program segment for a specified output given an incomplete program containing conditionals.
4. Debug a program containing an error involving conditional and iteration statements.
5. Predict the output of a program segment involving subroutines, functions, or methods.
6. Debug a program segment containing an error associated with subroutines, functions, or methods.
7. Predict the output of a program segment involving interacting objects.
8. Debug a program segment containing an error involving interacting objects.
9. Predict the output of a program segment involving parameters passed by value or reference.
10. Identify error types (i.e., syntax, runtime, logic).
11. Identify the purposes of internal and external program documentation.
12. Identify appropriate internal documentation for a group of program statements.
13. Identify appropriate preconditions or postconditions for given functions or methods.
14. Identify the strengths or weaknesses of object-oriented and procedural languages.

4 Knowledge of computer hardware

1. Identify the components of a computer system and their functions (i.e., input, output, processing, storage).
2. Distinguish between serial and data transfers.

3. Identify the major internal components of a microprocessor and their functions.
4. Identify the advantages and/or disadvantages of various storage media.

5 Knowledge of computer software

1. Identify the functions of a computer operating system.
2. Identify the advantages and/or disadvantages of programs that are compiled or interpreted.
3. Identify the features and functions of productivity software (e.g., word processing, spreadsheet, database, presentation, multimedia, Web authoring).

6 Knowledge of computer networking

1. Distinguish between various types of wired and wireless computer networks.
2. Identify the advantages and/or disadvantages of networked computing.
3. Identify the functions of the components of a network (e.g., servers, routers, switches, access points, workstations).
4. Identify features and functions of security software (e.g., firewalls, antivirus programs, filtering software, encryption).
5. Identify the advantages and/or disadvantages of different types of Internet connectivity.
6. Identify features and functions of digital communications (e.g., e-mail, instant messaging).
7. Identify features and functions of Hypertext Markup Language (HTML).
8. Identify features and functions of Web browsers.
9. Identify features and functions of search engines.

7 Knowledge of the social, environmental, ethical, and legal issues of computer technologies

1. Identify examples of appropriate use (e.g., software licensing, archival copying, fair use of copyrighted materials) and misuse (e.g., plagiarism, music and video piracy) of intellectual property.
2. Identify threats to privacy from centralized databases and commercial use of the Internet.
3. Identify examples of malicious interference with computer systems (e.g., viruses, worms, hacking, spam, spyware, denial-of-service attacks).

4. Identify the positive and negative impacts of computer technology.
5. Identify the roles and responsibilities of computer science professionals.

8 Knowledge of the history of computer technology

1. Identify important contributions of individuals or groups to the development of computer technology.
2. Identify generational milestones in the historical development of computer technology.

9 Knowledge of computer science pedagogy

1. Identify effective management strategies for teaching computer science (e.g., laboratory work, cooperative learning, electronic communications).
2. Identify appropriate instructional strategies for teaching computer science (e.g., case studies, role-playing, manipulatives, visualizations, simulations, modeling).
3. Identify appropriate assessment strategies for teaching computer science.
4. Identify appropriate accommodations and adaptations for students (e.g., students with exceptionalities, English language learners, students from various socioeconomic levels).